

1 What is claimed is:

2 1. In a data processing system including at least one client, a first file server coupled
3 to the client for data access of the client to at least one file in the first file server, and at
4 least a second file server coupled to the first file server for data access of the second file
5 server to the file in the first file server, the second file server being programmed with a
6 virus checker program, the virus checker program being executable by the second file
7 server to perform an anti-virus scan upon file data in random access memory of the
8 second file server, a method comprising:

9 the first file server responding to a request from the client for access to the file in
10 the first file server by determining that an anti-virus scan of the file should be performed,
11 and initiating the anti-virus scan of the file by sending to the second file server a request
12 for the anti-virus scan including a specification of the file, and then

13 the second file server responding to the request for the anti-virus scan by invoking
14 the virus checker program to perform an anti-virus scan of the specified file by obtaining
15 file data of the file from the first file server and storing the file data of the file into the
16 random access memory of the second file server and performing the anti-virus scan upon
17 the file data of the file in the random access memory.

18
19 2. The method as claimed in claim 1, wherein the first file server determines that the
20 anti-virus scan of the file should be performed when the client requests the first file
21 server to open the file and the first file server finds that the file has not been checked for
22 viruses.

1 3. The method as claimed in claim 1, wherein the first file server determines that the
2 anti-virus scan of the file should be performed when the client requests a file to be closed
3 after the client writes to the file.

4
5 4. The method as claimed in claim 1, wherein the first file server applies a filter to a
6 file extension of the file to determine that the anti-virus scan of the file should be
7 performed.

8
9 5. The method as claimed in claim 1, wherein the first file server blocks clients from
10 accessing the file from the time that the first file server determines that the anti-virus scan
11 of the file should be performed until the anti-virus scan is completed and fails to find a
12 virus in the file.

13
14 6. The method as claimed in claim 1, wherein the first file server determines that an
15 additional anti-virus scan of the file should not be performed in response to the access of
16 the file by the virus checker program.

17
18 7. The method as claimed in claim 1, wherein the first file server maintains in
19 nonvolatile memory an indication of files that have not been checked for viruses, an
20 indication of files that are in the process of being checked, and an indication of files that
21 have been found to contain viruses.

22

1 8. The method as claimed in claim 1, wherein the request for the anti-virus scan
2 including a specification of the file is an Open Network Computing Remote Procedure
3 Call.

4
5 9. The method as claimed in claim 1, wherein the second file server receives the
6 request for the anti-virus scan and indirectly invokes the virus checker program by
7 reporting a file access event to an operating system of the second file server, and the
8 operating system of the file server responds by invoking the virus checker program to
9 perform the anti-virus scan of the file.

10
11 10. The method as claimed in claim 1, wherein the operating system of the second file
12 server supports processes executing in a user mode and processes executing in a kernel
13 mode, and a server for virus checking executing in the user mode receives the request for
14 the anti-virus scan from the first file server and forwards the request to a virus checker
15 initiator driver executing in the kernel mode, and the virus checker initiator driver
16 executing in the kernel mode initiates a file access event, and the virus checker program
17 initiates the anti-virus scan of the file in response to the virus checker initiator driver
18 initiating the file access event.

19
20 11. The method as claimed in claim 10, wherein an input/output manager in the
21 operating system of the second file server receives a file access call from the virus
22 checker initiator driver, and responds by directing a report of the file access event to the
23 virus checker program.

1

2 12. The method as claimed in claim 1, wherein the data processing system includes at
3 least a third file server coupled to the first file server for data access of the third file
4 server to the file in the first file server, the third file server also being programmed with a
5 virus checker program that is executable by the third file server to perform an anti-virus
6 scan upon file data in random access memory of the third file server, wherein the first file
7 server performs a load balancing procedure to select one of at least the second file server
8 or the third file server to perform an anti-virus scan of the file when the first file server
9 determines that an anti-virus scan of the file should be performed.

10

11 13. A method of operating a network file server to initiate a virus scan upon a file
12 stored in the network file server, the network file server being coupled to at least one
13 client for access of the client to at least one file in the network file server, the network file
14 server being coupled to a plurality of secondary servers for access of the secondary
15 servers to the file stored in the network file server, the network file server including a
16 cached disk array and a plurality of data mover computers coupled to the data network
17 and coupled to the cached disk array for responding to client requests for data access to
18 storage in the cached disk array, each secondary server being programmed with a virus
19 checker program executable for performing an anti-virus scan upon file data in random
20 access memory of said each secondary server, a method comprising:

21 at least one of the data movers in the network file server responding to a request
22 from the client for access to the file in the network file server by applying a filter upon a
23 file extension of the file upon opening or closing of the file to determine that an anti-virus

1 scan of the file should be performed, and initiating the anti-virus scan of the file by
2 applying a load balancing procedure for selecting one of the secondary servers for
3 performing the anti-virus scan of the file, and sending to the selected secondary server a
4 request for the anti-virus scan including a specification of the file, and then
5 the selected secondary server responding to the request for the anti-virus scan by
6 invoking the virus checker program in the selected secondary server to perform an anti-
7 virus scan of the specified file by obtaining file data of the file from the network file
8 server and storing the file data of the file into the random access memory of the selected
9 secondary server and performing the anti-virus scan upon the file data of the file in the
10 random access memory of the selected secondary server.

11
12 14. The method as claimed in claim 13, wherein the file is an executable file, and the
13 network file server determines that the anti-virus scan of the file should be performed
14 when the client requests the network file server to open the file and the network file
15 server finds that the file has not been checked for viruses, and the network file server also
16 determines that the anti-virus scan of the file should be performed when the client
17 requests the file to be closed after the client writes to the file.

18
19 15. The method as claimed in claim 13, wherein the selected secondary server
20 receives the request for the anti-virus scan and indirectly invokes the virus checker
21 program by causing an operating system of the selected secondary server to invoke the
22 virus checker program in the selected secondary server to perform the anti-virus scan of
23 the file.

1

2 16. In a data network including a first server and a second server, the second server
3 being coupled by a data network to the first server for access of the second server to at
4 least one file stored in the first server, wherein the second server is programmed with an
5 operating system supporting processes executing in a user mode and processes executing
6 in a kernel mode, the operating system of the second server including an input/output
7 manager executing in the kernel mode, a method of operating the second server to
8 perform an anti-virus scan upon the file in the first server, said method comprising:

9 a server for virus checking executing in the second server in the user mode
10 receives from the network a request for the anti-virus scan upon the file, and then
11 the server for virus checking forwards the request to a virus checker initiator
12 driver executing in the second server in the kernel mode, and the virus checker initiator
13 driver responds to receipt of the request by sending a file access call to the input/output
14 manager, and then

15 the input/output manager responds to the file access call by directing a report of a
16 file access event to a virus checker program executing in the second server in the user
17 mode, and the virus checker program responds by obtaining file data from the file in the
18 first server and storing the file data in random access memory in the second server, and
19 performing an anti-virus scan upon the file data in the random access memory in the
20 second server.

21

22 17. In a data processing system including at least one client, at least one network file
23 server coupled to the client by a data network for access of the client to at least one file in

1 the network file server, and a plurality of NT file servers coupled to the network file
2 server by the data network for data access of the NT file servers to the file in the network
3 file server, the network file server including a cached disk array and a plurality of data
4 mover computers coupled to the data network and coupled to the cached disk array for
5 responding to client requests for data access to storage in the cached disk array, each of
6 the NT file servers being programmed with a virus checker program, the virus checker
7 program in each NT file server being executable by said each NT file server to perform
8 an anti-virus scan upon file data in random access memory of said each NT file server, a
9 method comprising:

10 a data mover in the network file server responding to a request from the client for
11 access to the file in the network file server by applying a filter upon a file extension of the
12 file upon opening or closing of the file to determine that an anti-virus scan of the file
13 should be performed, and initiating the anti-virus scan of the file by selecting a next one
14 of the NT file servers in round-robin fashion and sending to the selected NT file server a
15 request for the anti-virus scan including a specification of the file, and then

16 the selected NT file server responding to the request for the anti-virus scan by
17 invoking the virus checker program in the selected NT file server to perform an anti-virus
18 scan of the specified file by obtaining file data of the file from the network file server and
19 storing the file data of the file in the random access memory of the selected NT file server
20 and performing the anti-virus scan upon the file data of the file in the random access
21 memory of the selected NT file server.

22

1 18. The method as claimed in claim 17, wherein the file is an executable file and the
2 network file server determines that the anti-virus scan of the file should be performed
3 when the client requests the network file server to open the file and the network file
4 server finds that the file has not been checked for viruses, and the network file server also
5 determines that the anti-virus scan of the file should be performed when the client
6 requests the file to be closed after the client writes to the file.
7

8 19. The method as claimed in claim 17, wherein the operating system of the selected
9 NT file server supports processes executing in a user mode and processes executing in a
10 kernel mode, a server for virus checking executing in the user mode receives the request
11 for the anti-virus scan from the network file server and forwards the request to a virus
12 checker initiator driver executing in the kernel mode, the virus checker initiator driver
13 executing in the kernel mode sends a file access call to an input/output manager in the
14 operating system of the selected NT file server, the input/output manager responds to the
15 file access call by directing a report of a file access event to the virus checker program in
16 the selected NT file server, and the virus checker program in the selected NT file server
17 initiates the anti-virus scan of the file in response to receiving the report of the file access
18 event from the input/output manager.
19

20 20. A data processing system comprising:
21 at least one client;
22 a first file server coupled to the client for access of the client to at least one file in
23 the first file server; and

1 at least a second file server coupled to the first file server for data access of the
2 second file server to the file in the first file server, the second file server being
3 programmed with a virus checker program, the virus checker program being executable
4 by the second file server to perform an anti-virus scan upon file data in random access
5 memory of the second file server;

6 wherein the first file server is programmed to respond to a request from the client
7 for access to the file in the first file server by determining that an anti-virus scan of the
8 file should be performed, and initiating the anti-virus scan of the file by sending to the
9 second file server a request for the anti-virus scan including a specification of the file;
10 and

11 the second file server is programmed to respond to the request for the anti-virus
12 scan by invoking the virus checker program to perform an anti-virus scan of the specified
13 file by obtaining file data of the file from the first file server and storing the file data of
14 the file in the random access memory of the second file server and performing the anti-
15 virus scan upon the file data in the random access memory.

16
17 21. The data processing system as claimed in claim 20, wherein the first file server is
18 programmed to determine that the anti-virus scan of the file should be performed when
19 the client requests the first file server to open the file and the first file server finds that the
20 file has not been checked for viruses.

21

1 22. The data processing system as claimed in claim 20, wherein the first file server is
2 programmed to determine that the anti-virus scan of the file should be performed when
3 the client requests a file to be closed after the client writes to the file.
4

5 23. The data processing system as claimed in claim 20, wherein the first file server is
6 programmed to apply a filter to a file extension of the file to determine that the anti-virus
7 scan of the file should be performed.
8

9 24. The data processing system as claimed in claim 20, wherein the first file server is
10 programmed to block clients from accessing the file from the time that the first file server
11 determines that the anti-virus scan of the file should be performed until the anti-virus
12 scan is completed and fails to find a virus in the file.
13

14 25. The data processing system as claimed in claim 20, wherein the first file server is
15 programmed to determine that an additional anti-virus scan of the file should not be
16 performed in response to the access of the file by the virus checker program.
17

18 26. The data processing system as claimed in claim 20, wherein the first file server is
19 programmed to maintain in nonvolatile memory an indication of files that have not been
20 checked for viruses, an indication of files that are in the process of being checked, and an
21 indication of files that have been found to contain viruses.
22

1 27. The data processing system as claimed in claim 20, wherein the request for the
2 anti-virus scan including a specification of the file is an Open Network Computing
3 Remote Procedure Call.

4
5 28. The data processing system as claimed in claim 20, wherein the second file server
6 is programmed to receive the request for the anti-virus scan and indirectly invoke the
7 virus checker program by causing an operating system of the second file server to invoke
8 virus checker program to perform the anti-virus scan of the file.

9
10 29. The data processing system as claimed in claim 20, wherein the operating system
11 of the second file server supports processes executing in a user mode and processes
12 executing in a kernel mode, and the second file server includes a server for virus
13 checking that is executable in the user mode and a virus checker initiator driver that is
14 executable in the kernel mode, the server for virus checking being executable for
15 receiving the request for the anti-virus scan from the first file server and forwarding the
16 request to the virus checker initiator driver, and the virus checker initiator driver is
17 executable for causing the operating system reporting a file access event to the virus
18 checker program, and the virus checker program is executable for initiating the anti-virus
19 scan of the file in response to the report of the file access event.

20
21 30. The data processing system as claimed in claim 29, wherein the operating system
22 of the second file server includes an input/output manager that is executable in the kernel
23 mode for receiving a file access call from the virus checker initiator driver, and

1 responding to the file access call by directing the report of the file access event to the
2 virus checker program.

3
4 31. The data processing system as claimed in claim 20, which further includes at least
5 a third file server coupled to the first file server for data access of the third file server to
6 the file in the first file server, the third file server also being programmed with a virus
7 checker program that is executable by the third file server to perform an anti-virus scan
8 upon file data in random access memory of the third file server, and wherein the first file
9 server is programmed for performing a load balancing procedure to select one of at least
10 the second file server or the third file server to perform an anti-virus scan of the file when
11 the first file server determines that an anti-virus scan of the file should be performed.

12
13 32. A network file server adapted for coupling to at least one client for access of the
14 client to at least one file in the network file server, the network file server also being
15 adapted for coupling to a plurality of secondary servers for access of the secondary
16 servers to the file stored in the network file server, each secondary server being
17 programmed with a virus checker program executable for transferring file data from the
18 file in the network file server to random access memory in said each secondary server,
19 and performing an anti-virus scan upon the file data in the random access memory of said
20 each secondary server, the network file server comprising:

21 a cached disk array; and

22 a plurality of data mover computers coupled to the data network and coupled to
23 the cached disk array for responding to client requests for data access to storage in the

1 cached disk array, wherein at least one of the data movers is programmed to respond to a
2 request from the client for access to the file in the network file server by applying a filter
3 upon a file extension of the file upon opening or closing of the file to determine that an
4 anti-virus scan of the file should be performed, and initiating the anti-virus scan of the
5 file by applying a load balancing procedure for selecting one of the secondary servers for
6 performing the anti-virus scan of the file, and sending to the selected secondary server a
7 request for the anti-virus scan including a specification of the file.

8
9 33. The data processing system as claimed in claim 32, wherein the file is executable
10 and the network file server is programmed to determine that an anti-virus scan of the file
11 should be performed when the client requests the network file server to open the file and
12 the network file server finds that the file has not been checked for viruses, and the
13 network file server is also programmed to determine that the anti-virus scan of the file
14 should be performed when the client requests the file to be closed after the client writes to
15 the file.

16
17 34. A secondary server adapted for coupling to a primary server in a data network for
18 access to data in files in the primary server, wherein the secondary server is programmed
19 with an operating system supporting processes executing in a user mode and processes
20 executing in a kernel mode, the operating system including an input/output manager
21 executable in the kernel mode, wherein the secondary server is further programmed with:
22 a server for virus checking executable in the user mode;
23 a virus checking driver executable in the kernel mode; and

1 a virus checker program executable in the user mode;
2 wherein the server for virus checking is executable for receiving from the network
3 a request for an anti-virus scan upon a specified file in the primary server, and for
4 forwarding the request to the virus checker initiator driver;
5 wherein the virus checker initiator driver is executable for responding to receipt of
6 the request from the server for virus checking by sending a file access call upon the
7 specified file to the input/output manager;
8 wherein the input/output manager is executable for responding to the file access
9 call by directing a report of a file access event upon the specified file to the virus checker
10 program; and
11 wherein the virus checker program is executable for responding to the report of
12 the file access event by transferring file data from the specified file in the primary server
13 to random access memory in the secondary server, and performing an anti-virus scan
14 upon the file data in the random access memory in the secondary server.

15
16 35. A data processing system comprising:
17 at least one client;
18 at least one network file server coupled to the client by a data network for access
19 of the client to at least one file in the network file server; and
20 a plurality of NT file servers coupled to the network file server by the data
21 network for data access of the NT file servers to the file in the network file server;

1 wherein the network file server includes a cached disk array and a plurality of
2 data mover computers coupled to the data network and coupled to the cached disk array
3 for responding to client requests for data access to storage in the cached disk array;

4 wherein each of the NT file servers is programmed with a virus checker program,
5 the virus checker program being executable by said each NT file server to perform an
6 anti-virus scan upon file data in random access memory of said each NT file server;

7 wherein at least one data mover in the network file server is programmed for
8 responding to a request from the client for access to the file in the network file server by
9 applying a filter upon a file extension of the file upon opening or closing of the file to
10 determine that an anti-virus scan of the file should be performed, and initiating the anti-
11 virus scan of the file by selecting a next one of the NT file servers in round-robin fashion
12 and sending to the selected NT file server a request for the anti-virus scan including a
13 specification of the file, and

14 wherein each NT file server is programmed to respond to the request for the anti-
15 virus scan by invoking the virus checker program in said each NT file server to perform
16 an anti-virus scan of the specified file by obtaining file data of the specified file from the
17 network file server and storing the file data of the specified file in the random access
18 memory of said each NT file server and performing the anti-virus scan upon the file data
19 of the specified file in the random access memory of said each NT file server.

20
21 36. The data processing system as claimed in claim 35, wherein the file is an
22 executable file, and the network file server is programmed for determining that the anti-
23 virus scan of the file should be performed when the client requests the network file server

1 to open the file and the network file server finds that the file has not been checked for
2 viruses, and the network file server also determines that the anti-virus scan of the file
3 should be performed when the client requests the file to be closed after the client writes to
4 the file.

5
6 37. The data processing system as claimed in claim 35, wherein the selected NT file
7 server is programmed with:

8 an operating system supporting processes executing in a user mode and processes
9 executing in a kernel mode, the operating system including an input/output manager
10 executable in the kernel mode;

11 a server for virus checking executable in the user mode; and

12 a virus checker initiator driver executable in the kernel mode;

13 wherein the server for virus checking is executable for receiving the request for
14 the anti-virus scan from the network file server and forwarding the request to the virus
15 checker initiator driver;

16 wherein the virus checker initiator driver is executable for responding to the
17 request from the server for virus checking by sending a file access call to the input/output
18 manager;

19 wherein the input/output manager is executable for responding to the file access
20 call by directing a report of a file access event to the virus checker program in the
21 selected NT file server; and

1 wherein the virus checker program in the selected NT file server is executable for
2 performing the anti-virus scan of the file in response to receiving the report of the file
3 access event from the input/output manager.
4

5 38. A program storage device containing a program executable by a network file
6 server, the network file server being adapted for coupling to at least one client for access
7 of the client to at least one file in the network file server, the network file server also
8 being adapted for coupling to a plurality of secondary servers for access of the secondary
9 servers to the file stored in the network file server, each secondary server being
10 programmed with a virus checker program executable for transferring file data from the
11 file in the network file server to random access memory in said each secondary server,
12 and performing an anti-virus scan upon the file data in the random access memory of said
13 each secondary server, the program contained in the program storage device being
14 executable by the network file server for responding to a request from the client for
15 access to the file in the network file server by applying a filter upon a file extension of the
16 file upon opening or closing of the file to determine that an anti-virus scan of the file
17 should be performed, and initiating the anti-virus scan of the file by applying a load
18 balancing procedure for selecting one of the secondary servers for performing the anti-
19 virus scan of the file, and sending to the selected secondary server a request for the anti-
20 virus scan including a specification of the file.
21

22 39. The program storage device as claimed in claim 38, wherein the file is an
23 executable file, and the program contained in the program storage device is also

1 executable by the network file server for determining that an anti-virus scan of the file
2 should be performed when the client requests the network file server to open the file and
3 the network file server finds that the file has not been checked for viruses, and the
4 program contained in the program storage device is also executable by the network file
5 server for determining that the anti-virus scan of the file should be performed when the
6 client requests the file to be closed after the client writes to the file.

7
8 40. A program storage device containing a program executable by a secondary server,
9 the secondary server being adapted for coupling to a primary server in a data network for
10 access to data in files in the primary server, wherein the secondary server is
11 programmable with an operating system supporting processes executing in a user mode
12 and processes executing in a kernel mode, the operating system including an input/output
13 manager executable in the kernel mode, and the secondary server is also programmable
14 with a virus checker program for performing an anti-virus scan upon file data in response
15 to a file opening event being reported to the input/output manager, wherein the program
16 contained in the program storage device includes:

17 a server for virus checking executable in the user mode; and

18 a virus checking driver executable in the kernel mode;

19 wherein the server for virus checking is executable for receiving from the network
20 a request for the anti-virus scan upon a specified file in the primary server, and for
21 forwarding the request to the virus checker initiator driver; and

22 wherein the virus checker initiator driver is executable for responding to receipt of
23 the request from the server for virus checking by sending a file access call upon the

